

(FILE 'USPAT' ENTERED AT 17:40:23 ON 01 SEP 1999)
ACTIVATE S09035995/L

L1 (28280)SEA (RESERV? OR ESTABLISH?) (3A) (CHANNEL? OR PATH?)
L2 (1524)SEA CONVERT? (3A) (DATA FORMAT? OR PROTOCOL?)
L3 (191)SEA L1 AND L2
L4 (42661)SEA ENCOD? AND DECOD?
L5 (60)SEA L3 AND L4
L6 (227904)SEA (ATTRIBUT? OR TYPE? OR CHARACTERISTIC?) (3A) (NODE# OR
CO
B# MPUTER# OR EQUIPMENT OR UNIT# OR SYSTEM# OR NETWORK# OR HU
OR RESOURCE#)
L7 (43)SEA L5 AND L6
L8 (246)SEA MEMORY AND BROADCAST AND ATM AND IP
L9 (5)SEA L7 AND L8
L10 (43)SEA L5 AND L6
L11 (0)SEA TRANSLAT? (3A) (DATA FORMAT? OR PROTOCOL?) AND L1 AND
L4
AND L5 AND L6 NOT L7
L12 (0)SEA TRANSLAT? (3A) (DATA? OR CHANNEL# OR PATH? OR PROTOCOL
?)
AND L1 AND L4 AND L5 AND L6 NOT L7
L13 (74)SEA IEEE 1394

L14 1 SEA ROUTER (2A) (MULTIPLE? OR PLURAL?) (2A) PROTOCOL?

ABSTRACT:

Method, . . . and apparatus are described for automatically receiving, at an intermediate processing location, data from a wide variety of remote sources, **identifying** the format of the data, translating the data to a common file format, sending the data to a recipient in. . .

SUMMARY:

BSUM(12)

The . . . comprises a system capable of automatically receiving, at an intermediate processing location, data from a wide variety of remote sources, **identifying** the format of the data, translating the data to a common format, sending the data to a recipient in an. . .

DETDESC:

DETD(8)

Destination **Address**--Information that tells the system to whom a data file is to be sent, and therefore, it which Outgoing Data Box. . .

DETDESC:

DETD(33)

Subscriber ID--The unique **identification** key assigned to each subscriber.

DETDESC:

DETD(36)

Subscriber Table--A database file containing subscriber information such as Subscriber ID number, Subscriber Name, Job Number, Translation Information, Subscriber Destination **Address**, and subscriber Filename Information.

DETDESC:

DETD(57)

A . . . these data can be divided into general categories, including, but not limited to ASCII, ANSI X.12, EDIFACT, Binary Files with **Headers**, graphics files, sound files, video files, and alternate compression method files. Such general categories may be further subdivided into an. . . Data File 115 is generically referred to herein as the Provider's Data Format. Regardless of the format used, a Destination **Address** 117, which describes who the data is for, accompanies the actual data within Provider Data File 115.

DETDESC:

DETD(73)

In . . . pertaining to a specific Data Subscriber into a database application such as dbase (Borland, Scotts Valley, Calif.). Adding a

Destination **Address** to such a data file is performed by a variety of means, including simply typing in an alphanumeric string. . . . into a certain field of their file of cleared bank checks. In step 120, Provider Data File 115 with Destination **Address** 117 is loaded into Provider Section 10 computer, into Outgoing Data Box 128. Here, this particular process halts, and does. . . . in detail in the discussion of FIG. 5, steps 235 and 245. The basic problem that all of these methods **address** is that the quality of data ultimately delivered to Data Subscribers 33 must be ensured. Not having safeguards would be. . . .

DETDESC:

DETD(95)

Validation, 235 is performed in any of a variety of ways. In one embodiment, Main Processing Section 20 employs the caller **identification** method to determine whether the caller is valid. By this caller ID method, the telephone number of any party attempting. . . .

DETDESC:

DETD(110)

In PreProcessor 363 is needed, based upon which MPS Incoming Data Box contains a Provider Data File 115, or by Destination **Address** 117 residing inside of Provider Data File 115.

DETDESC:

DETD(117)

In step 305, PreProcessor 363 gets Subscriber ID 308 located in Provider Data File 115 **header**. This is done by looking in a given byte location for a sequence of bytes making up Subscriber ID 302,. . . .

DETDESC:

DETD(126)

Returning 354 distributes MidFormat File 342 from Work Area 363 to the appropriate MPS Outgoing Data Boxes 371, using Subscriber Destination **Address** 117 as a guide 371A, 371B and 371C are examples of different Subscriber Outgoing Data Boxes 371, each dedicated to. . . .

DETDESC:

DETD(135)

Provider 1). The purpose of this database is to store relevant information about the customers such as the customer's name, customer's **identification, address, etc.**

DETDESC:

DETD(138)

In an alternative embodiment, PrepServer Software 354 detects the appropriate destination from MidFormat File 342 characteristics other than a Subscriber Destination **Address** 117. Such characteristics include but are not limited to the file name, the file size, the file **header**, and the file trailer. By such cues, PrepServer Software 354, in one embodiment, determines the target MPS Outgoing Data Box. . . .

DETDESC:

DETD(162)

FIG. . . . initiates contact with Main Processing Section 20 with telecommunications link 21, for example by modem. Subsequently, Subscriber Section 30 is **identified** by Main Processing Section 20, and assessed as to whether the link is with a valid subscriber. Unless the call. . . .

DETD(DESC:

DETD(163)

In one embodiment, Main Processing Section 20 employs the caller **identification** method. By this caller ID method, the telephone number of any party attempting to establish connection with Main Processing Section. . . .

DETD(DESC:

DETD(207)

When . . . perform the following steps: accept an input file in the form of a fax or scanned image of a document, **identify** the type of document represented, align the image, locate each field of information, extract each field, segment or cut between. . . .

DETD(DESC:

DETD(208)

FIG. . . . via a telecommunication link carrying Fax Bitstream 710 between the fax machine and Fax/Modem 780. Once an incoming fax is **identified**, the computer(s) at Main Processing Section 20 capture each incoming fax page as a unique Fax Bitmap 725 image file. . . . directory. Meanwhile, Fax PrepServer A 720 continuously scans Fax Queue 715 for Fax Bitmap 725 image files. Once PrepServer A **identifies** Fax Bitmap 725 image files in the Fax Queue 715, Fax PrepServer A 720 takes the first Fax Bitmap 725. . . .

DETD(DESC:

DETD(209)

Fax . . . instructions. Forms Processing Software 742 perform the following steps: accepts input files in the form of TIFF images of documents, **identifies** the type of document represented (by means such as reading **identifying** information on the image of a cover page which precedes the transmission of the actual data, or by the recognition of **identifying** information on the images of actual data pages, such as an account number, name, or a specific form layout), align. . . .

DETD(DESC:

DETD(217)

ID . . .
ID 6 Alphanumeric *Drives internal
AR billing system
(Platinum AR)
Customer Name 40 Alphanumeric
Contact Person 30 Alphanumeric
Address 1 30 Alphanumeric
Address 2 30 Alphanumeric
City 40 Alphanumeric
State 2 Alphanumeric

Country 15 Alphanumeric
Zip Code 10 Alphanumeric
Voice Number 10 Alphanumeric. . .

DETDESC:

DETD(219)

SIZE;

Data Transmission Time

8 Numeric

INCLUDED FOR

Target Data Module

2 Alphanumeric

CONSISTENCY.

Number of **Headers**

8 Numeric

Number of Line Items

8 Numeric

Time (A8) is "HH:MM:SS"

CLAIMS:

CLMS(2)

2. . . . said step (A) of transmitting, further includes, prior to said step of monitoring, the steps of:
appending a subscriber destination **address** to said provider data file uniquely representative of a specific subscriber that the data in said particular data file is intended to ultimately reach; and loading said provider data file with said appended destination **address** into an outgoing data box associated with said provider within said provider section.

CLAIMS:

CLMS(4)

4. The method in claim 1, wherein said provider data file format is a digital image data bit stream **encoding** a printed document.

CLAIMS:

CLMS(9)

9. . . . processing procedure includes the steps of:
accepting an input file in the form of a graphical image of a document; **identifying** the type of document represented from among a plurality of known possible document types; aligning the graphical image with known. . . .

CLAIMS:

CLMS(23)

23. The method in claim 20, wherein said provider data file format is a digital image data bit stream **encoding** a printed document.

CLAIMS:

CLMS(28)

28. . . . processing procedure includes the steps of:

accepting an input **image** in the form of a graphical **image** of a document;
identifying the type of document represented from among a plurality
of known possible document types; aligning the graphical image with

SUMMARY:

BSUM(22)

The inventions relate to **networks**. **Networks** of some **type** have existed for many years to connect sending devices to receiving devices, and the early networks were telephone related. Coexisting. . .

SUMMARY:

BSUM(28)

Other . . . by the present inventions include complete and incomplete Hypercubes and Benes networks which are in turn derived from Clos. Hypercube **type networks** are described in "Incomplete Hypercubes", H. Katseff; IEEE Transactions on Computers, Vol. 37, No. 5, May 1988, 0018-9340/88/0500-0604, pp. 604-608;. . .

SUMMARY:

BSUM(29)

Shuffle . . . is a particular kind of multi-stage network with an interconnection pattern which is described as "Perfect Shuffle". This kind of **network** has better scalability **characteristics** than some other **networks**, and sometimes can achieve good performance. Another application of a shuffle network is for SIMD, as described in "On the. . .

SUMMARY:

BSUM(33)

Other . . . in a second mode wherein said one switching plane and said other switching planes are used for data transfer. This **type** of switching **system** may be called a collision crossbar. It requires at least 10 clocks to set up every stage, is synchronous, and. . .

SUMMARY:

BSUM(42)

The . . . interface to the parallel switch. A first MULTI-MEDIA SERIAL LINE SWITCHING ADAPTER, disclosed herein, receives either serial optical or electrical interfaces/**protocols** and **converts** them into the parallel electrical interfaces/protocols required by the parallel switch. The converted serial data is routed to the selected. . .

SUMMARY:

BSUM(43)

The . . . is personalized by the PROM to understand the protocol generated by that node. The first MULTI-MEDIA SERIAL LINE SWITCHING ADAPTER **converts** the first serial **protocol** to the parallel protocol of the ALLNODE switch and forwards the converted message to the parallel network. The message is. . .

DRAWING DESC:

DRWD(12)

FIG. 11 illustrates the typical method of selecting and **establishing** a transmission **path** through a network comprised of the invention switching apparatus for the purpose of sending data from one node to another.

DRAWING DESC:

DRWD(33)

FIGS. . . . diagram, which shows a typical example of the network send function in regards to sending a message to a previously **established path** in the parallel switch network.

DETDESC:

DETD(4)

The . . . it is finished using a given switch path. Thus, the latency of the central matrix controller approach in regards to **establishing** and breaking switch **paths** is very poor. In existing products, this type of approach has been adequate to connect DASD's, Direct Access Storage Devices. . . .

DETDESC:

DETD(10)

Referring . . . network 30. For example, node 1A could send a message using serial data channel 40A to MMSA 20A. MMSA 20A **converts** the serial message **protocol** to a parallel message protocol and sends it to switch interface 46A. MMSA 20A receives destination identification data (DID) as. . . 20A through network 30 to MMSA 20N. MMSA 20N receives the parallel message protocol over switch interface 46N. MMSA 20N **converts** the parallel message **protocol** to the designated serial message protocol and sends it to node 1N over serial channel 40N. Node 1A sends its. . . response message it generates. Node 1N sends the response message over serial data channel 40N to MMSA 20N. MMSA 20N **converts** the serial message **protocol** to a parallel message protocol and sends it to switch interface 46N. MMSA 20N uses the DID it receives to. . . MMSA 20N through network 30 to MMSA 20A. MMSA 20A receives the parallel message over switch interface 46A. MMSA 20A **converts** the parallel message **protocol** to the designated serial message protocol and sends it to node 1A over serial channel 40A. In similar fashion any. . .

DETDESC:

DETD(11)

Referring . . . 20 (MMSA) and the switch network 30 design is demonstrated. The flexibility allows the present invention to support a nodal **system** comprised of various **types** of interfaces. The MMSA can be personalized to support the any one of a number of standard and proprietary serial. . .

DETDESC:

DETD(14)

A . . . illustrates that any node of the system, such as node 4, can

serve as a bridge to another network, whereby **networks** the same **type** as **network 30** (ALL-Node Switch Networks) can be interconnected to each other through network bridge nodes, like node 4. It is also possible to interconnection to network 30, other networks which are of a different **type** than **network 30**, such as telephone networks, wireless **networks**, or other **types** of multi-stage or crossbar networks through network bridge nodes, like node 4. FIG. 2 shows that bridge nodes, like node . . .

DETDESC:

DETD(16)

Referring . . . when it is finished using a given switch path. Thus, the latency of the matrix switch approach in regards to **establishing** and breaking switch **paths** is very poor. Numbers quoted for the latency of existing matrix switch products usually are in the range of 10. . .

DETDESC:

DETD(17)

In contrast, the present invention applies to low latency fully parallel switch networks, also shown in FIG. 4. In this **type** of **network** there is no shared matrix controller, but the switch fabric and path set-up and breakdown control is provided by the. . .

DETDESC:

DETD(34)

The . . . two clock cycles of serial data over interface 31 from serial registers 54 to switching apparatus 14) to select and **establish** a connection **path** through the switching apparatus 14. The example in FIG. 9 illustrates via dashed lines, the switching apparatus establishing a temporary. . .

DETDESC:

DETD(37)

In . . . of switching apparatus 10 to which to connect. The method of path selection recommended here is one out of N **encoding** with a return to zero (called a DEAD FIELD).

DETDESC:

DETD(41)

In . . . all zeroes (a DEAD FIELD) as shown in FIG. 11. Thus, by clock time 4, switches 10A and 10F have **established** a connection **path** for transferring data directly from Node 1 to Node 7. Up to clock time 5, Node 7 sees nothing but. . . switch 10F. The protocol of the actual data being transmitted can be any of the normal formats such as manchester **encoded**, 8/10 bit **encoding** with preamble, etc. However, the preferred embodiment, as shown in FIG. 11 is an all ones synchronization field at time. . .

DETDESC:

DETD(50)

Referring . . . preferred embodiment 64 nodes because most algorithms which have needed parallel implementation would appear to be serviced

adequately by 64 nodes, preferably RS/6000 type nodes. This number should not be considered limiting, as 512 or 320 nodes can be made into a network with the . . .

DETDESC:

DETD(66)

Referring . . . and trailer interpreted by block 140. Then, the data message is translated into a data message having the parallel switch **protocol**. The **converted** message is stored in the Switch Message Send Buffer 160 until it is received in full, then the Switch Message. .

DETDESC:

DETD(76)

As . . . 202 of FIG. 17. The processing involves converting them from 10-bit characters to 8-bit bytes by the 10 to 8-bit **decoder** 208. For the preferred serial protocol, the only conversion required for the link header and message data is the 10. . .

DETDESC:

DETD(87)

Referring . . . to IT15 time period and thereafter. The combination of latch 440 being set and latch 442 not being set is **decoded** in gate 466 and defines the second IT12 to IT15 time period when Header Word 1 is written into buffer. . .

DETDESC:

DETD(89)

Referring . . . gate 507 from a combination of the high order 16 bits from register 520 (DID) and 16 bits of the **decoded** network command from block 542 of FIG. 26. Message Data Words are gated to buffer 160 from 4-byte Data Register. . .

DETDESC:

DETD(93)

Referring . . . destination node over network 30. The timing for the Network Message Send function is shown in FIG. 31 including the **establishment** of the network **path** and the sending of the valid message over the network path. When the BUFFER EMPTY signal from block 238 (FIG. . . .

DETDESC:

DETD(95)

Referring . . . preferred network 30 embodiment has two network stages as shown by FIG. 13, and requires two stage routing bytes to **establish a path** through network 30--a stage 1 routing byte which is constructed at time ST1 and a stage 2 routing byte that is constructed at ST3 time, as shown in FIG. 31. 3-bit selector 604 and 3-to-8 **decoder** 605 combine to construct the stage 1 and 2 routing bytes from the destination node bytes stored in register 600. . . ST1, selector 604 selects the low order 3 bits of byte 1 of register 600 to be gated through to **decoder** 605 which converts the 3-bits to an 8 bit code that becomes the stage 1 routing byte, which is transmitted. . .

selector 604 selects the next high order 3 bits of byte 1 of register 600 to be gated through **decoder** 605 which converts the 3 bits to an 8 bit code that becomes the stage 2 routing byte, which is transmitted. . . is used to enable blocks 604 and 605 only at ST1 or ST3 times, so that for all other times **decoder** 605 passes all zeroes to gate 618; this causes the all zeroes dead fields to be generated during times ST2. . .

DETDESC:

DETD(96)

Also . . . which contain the command field are loaded into register 602, as well as register 600. The Command Register 602 is **decoded** by block 608 into 5 different Local Link Command types. This **decode** is based on the bit definition shown in block 554 of FIG. 27, where bits 21 to 23 define the . . . Command) is set, the command is designated to be executed by block 180, and bits 20 to 23 should be **decoded** to define the specific command type. If bit 16 (Remote Link Command) is set, the command is designated to be . . . bytes and their associated dead fields usually generated by times ST1 to ST4. This is possible because the network 30 **path** connections are already **established** and the portion of the network header which usually performs this function can be skipped.

DETDESC:

DETD(99)

Referring . . . (Local Link Command) is latched in register 602 (FIG. 29) and sent to gate 626 and used to enable the **decoding** of the local commands to be executed by block 180. Local Link commands are not sent to network 30, as . . . transmission of the Local Link Message to network 30. Instead, the Local Write Node ID command, as generated by command **decoder** 608, goes to register 650 through gate 660 and causes a data message word to be written into register 650. . . the network port, a Switch Reset, and other network controls. The Local Write Control Reg command as generated by command **decoder** 608 goes to register 652 through gate 664 and causes a data message word to be written into register 652. . .

DETDESC:

DETD(105)

FIG. . . . command being received is a Remote Link Command as specified by bit 18 of register 554 of FIG. 27 and **decoded** by register 775 and gate 772). When block 170 is commanded to the IGNORE MODE, it never issues OUTX-ACCEPT, OUTX-REJECT, . . . driven active, if the inverse Remote Link Command as specified by bit 18 of register 554 of FIG. 27 and **decoded** by register 775 and gate 754 is active. The END OF MESSAGE signal 752 goes to buffer 150 and causes. . .

DETDESC:

DETD(107)

Referring . . . used to over-ride some normal valid message receiving sequences. In addition, other bits 20 to 23 of register 775 are **decoded** in block 776 to define the individual Remote Link Commands to be executed by block 170. Remote Link commands are. . . Link command from being stored successfully into buffer 150. Instead, the Remote Write Node ID command, as generated by command **decoder** 776, goes to register 650 through gate 660 and causes a data message word to be written into register 650. . . the network port, a Switch Reset, and other network controls. The Remote Write Control Reg command as generated

by command **decoder** 704 goes to register 652 through gate 664 and causes a data message word to be written into register 652. . .

DETDESC:

DETD(108)

Referring . . . serial message. The serial bit stream sent to module 120 or 122 and is converted to 10-bit parallel data by **encoder** 704 and transmitted through register 702. Register 702 sends all valid messages which are converted into the 10-bit characters of. . .

DETDESC:

DETD(111)

Referring . . . of FIG. 39), and brings together the various portions of the serial message by feeding them to register 702 through **encoder** 704 for transmission to the destination node. The eleventh bit selected by MUX 972 is of the Enable bit for. . .

DETDESC:

DETD(115)

Referring . . . to TT11 cause the associated 10-bit characters to be read from register block 715, sent to Transmit Assembly MUX 708, **encoder** 704, register 702, and finally to the serial transmitter module 120 or 122. The processing performed by block 130 involves. . . DID signal goes to gate 982 of FIG. 44, where it is time multiplexed during times TT0 to TT3 to **encoder** 704 to be converted to 10-bit characters before being sent to register 702 through OR MUX 708. The SID register. . . SID signal goes to gate 982 of FIG. 44, where it is time multiplexed during times TT4 to TT7 to **encoder** 704 to be converted to 10-bit characters before being sent to register 702 through OR MUX 708. The Network Command. . . Header Word 1 data received from buffer 150 at LOAD HEADER WORD 1 time as shown in FIG. 42. The **Decode** Link Command block 966 uses the data stored in register 954 to generate up to 4 LCNTL bytes as controlled. . . LCNTL signal goes to gate 982 of FIG. 44, where it is time multiplexed during times TT0 to TT3 to **encoder** 704 to be converted to 10-bit characters before being sent to register 702 through OR MUX 708. The Message Data. . .

DETDESC:

DETD(116)

Referring . . . TT15 time goes through gate 978 to gate 982, where it is time multiplexed during times TT12 to TT15 to **encoder** 704 to be converted to 10-bit data characters before being sent to register 702 through OR MUX 708. After the. . .

DETDESC:

DETD(120)

Referring . . . TRANS CRC signal to gate 982 of FIG. 44, where it is time multiplexed during times TT16 to TT19 to **encoder** 704 to be converted to 10-bit characters before being sent to register 702 through OR MUX 708. The preferred embodiment. . . 5 generators are active simultaneously, receive byte serial data simultaneously from MUX 982 (FIG. 44) as it is sent to **encoder** 704, and generate 5 different types of common CRC fields simultaneously. During the CRC times TT16 to TT17, the selected. . .

DETDESC:

DETD(121)

Referring . . . of FIG. 39), which brings together the various portions of the serial message by feeding them to register 702 through **encoder** 704 for transmission to the destination node. The eleventh bit selected by MUX 972 is of the Enable bit for. . .

DETDESC:

DETD(122)

Likewise, . . . to gate 708, which brings together the various portions of the serial message by feeding them to register 702 through **encoder** 704 for transmission to the destination node. The eleventh bit selected by MUX 971 is not used, as the ABF. . .

DETDESC:

DETD(143)

The . . . into the PC or workstation memory from one of the various serial transfer media. Serial to parallel emulator software would **convert** the serial message **protocol** to the ALLNODE Switch Network parallel protocol. In the preferred embodiment the PC or workstation would relay the converted message. . . workstation, which would receive the switch parallel protocol message into the PC or workstation memory. Then, the emulator software would **convert** the parallel message **protocol** to the a serial protocol, and relay the message to the appropriate serial interface.

DETDESC:

DETD(148)

With . . . of them can interact. This is because the communication time is infinitesimally small. The "traffic cop" is eliminated and parallel **paths** can be **established** to interconnect multiple processors. In general, in a parallel processing system when there are multiple processors, the processors can be. . .

DETDESC:

DETD(152)

While interconnection **networks** of some **type** have been used in computer systems, there is a need to provide a way for systems to be open, to. . .

DETDESC:

DETD(153)

In . . . now to get excellent performance from short messages as well as long messages. The present system can replace the older **type** long message **systems** which were basically I/O switches and allowed devices like DASD's to send very long messages. This present system can still. . .

CLAIMS:

CLMS(1)

What . . . is:

1. A computer system comprising:
a plurality of processor nodes each including means for transmitting in standard 8-to-10 bit **encoded** serial format a connection request and a data message;
an interconnection network, said interconnection network including a plurality of bufferless switching. . . the discrete control lines control the data message flowing through the switch network and are derived from special 8-to-10 bit **encoded** characters embedded within the standard serial data message; and
each of the adapters further including a second means for receiving said. . . means for recovering the data message from the switch network, converting the data and discrete control lines to 8-to-10 bit **encoding** and embedding said controls with said data to couple network output port to standard serial protocol.

CLAIMS:

CLMS (8)

8. . . . comprising:
a plurality of processor nodes, each including means for transmitting a connection request and information in standard 8-to-10 bit **encoded** serial format and a means for receiving information;
an interconnection network, said interconnection network including a plurality of bufferless switching apparatuses. . . the discrete control lines control the data message flowing through the switch network and are derived from special 8-to-10 bit **encoded** characters embedded within the standard serial data message; and
each of the adapters further including a second means for receiving said. . . means for recovering the data message from the switch network, converting the data and discrete control lines to 8-to-10 bit **encoding** and embedding said controls with said data to couple

ABSTRACT:

A . . . token bus, token ring, and fiber distributed data interface (FDDI), can be implemented using the generic channel architecture and its **characteristics** providing respective **network** functions. The architecture also provides a digital collision detection method and provides information necessary for precise network statistics monitoring. The . . .

SUMMARY:

BSUM(25)

3. Attempts to address these limitations in the prior art lead to higher costs: additional backplane buses or complex **protocol converters** (e.g. network bridges).

SUMMARY:

BSUM(31)

It is an object of the invention to provide a system and an arrangement for **networks** of all media **types**, including twisted pair, fiber optic, thin and thick coaxial cable and others by employing a concentrator which is modular and. . .

SUMMARY:

BSUM(36)

Control Module: A central module that performs functions uniquely related to a **hub**. Example of these **type** of cards are the repeater module in the Multiconnect.TM., and Re-timing module in the System 3000.TM.. The functions associated with. . .

SUMMARY:

BSUM(39)

Bridge Module: A module that implements any type of store-and-forward-function for any purpose. It either **converts** one **protocol** to another, or filters (receive all transmissions from one port, and selectively transmits to another port, or both).

SUMMARY:

BSUM(62)

The . . . may observe an inter-packet gap of a few bit times (a bit time is defined as 100 nano-seconds). When this **type** of **network** scenario occurs, the Ethernet.TM. controller IC may not receive the subsequent packet following the gap and ignore its activity. An. . .

DETDDESC:

DETD(10)

The bridging and routing modules 122, 124 are store-and-forward devices that either **convert one protocol** to another, filter (receives all transmissions from one port, and selectively transmits to another port), or both.

DETDESC:

DETD(25)

FIG. 4 is a module switching schematic which shows an **encoded** signal representing the selected active channel. Four possible combinations of the signal can be selected for the generic channels A, . . .

DETDESC:

DETD(31)

The input signals ENCHNLB<1:0> represent the selected channel (one of Channel 0, 1, 2 or Isolated) in binary **encoded** form. The **decoder** U66 translates these **encoded** signals to generate ENCHNLO3 and ENISOLATE output signals. These output signals enable respective channels for LAN communications. The modules switch channels. . .

DETDESC:

DETD(40)

2. . . . are developed, a method of supporting the new protocols on the generic data channel can be designed and included. Costly **protocol converters** are not required.

DETDESC:

DETD(118)

The signals BID<4:0> are a uniquely **encoded** Slot-ID signal per module such that the slot 0 of the concentrator has the binary **encoding** of 00000 and the slot 15 of the concentrator has the **encoding** of 01111. The signals MONADD<8:4> represent the portion of the active module address that corresponds to the Slot-ID. When a. . .

DETDESC:

DETD(133)

FIG. . . . of the state machine described in FIGS. 13 and 14. The programmable logic device U11, among other independent functions, provides the **decoder** function to translate ENCHNLB<1:0> to ENCHNL<2:0>. For the single port implementation, such as shown here for a Management module, the. . .

DETDESC:

DETD(192)

The . . . uses these lines to determine which of the 17 backplane lines to receive its data from. As with the transmit **decoder**, this function can be overridden by network management. The operation of this element is basically as follows:

CLAIMS:

CLMS(1)

What . . .

one or more ports, each of said plurality of media modules being provided for one or more physical local area **network** media **type** such as fiber optic medium, coaxial cable or twisted pair cable and each of said ports being provided for a . . .

CLAIMS:

CLMS (9)

9. . . .
one or more ports, each of said plurality of media modules being provided for one or more physical local area **network** media **type** such as fiber optic medium, coaxial cable or twisted pair cable and ports of each media module being provided for. . .

CLAIMS:

CLMS (15)

15. . . . accordance with claim 13, wherein:
said management means controls and monitors said Token Ring protocol on said plurality of communication **paths**, said management means **establishes** a logical ring based on said unique slot-ID.

US PAT NO: 5,208,811 [IMAGE AVAILABLE] L20: 4 of 4
TITLE: Interconnection system and method for **heterogeneous networks**

SUMMARY:

BSUM(2)

The present invention relates to an interconnection system and method for **heterogeneous networks**, and in particular, to an interconnection system and method for **heterogeneous networks** in which a terminal of a local area network (LAN) and a terminal of an integrated services digital network (ISDN). . .

SUMMARY:

BSUM(3)

In **heterogeneous network** systems, in order to allow communications to be accomplished between two terminals, namely, between an LAN terminal connected to an. . .

SUMMARY:

BSUM(9)

Namely, . . . to be provided with three function as shown in FIG. 2C, namely, the functions of the bridge 110, an LAN/ISDN **protocol converter** 130 (for an address translation and a route or routing control), and the ISDN inter-connector 120.

SUMMARY:

BSUM(10)

The address translation and routing control unit, namely, the LAN/ISDN **protocol converter** 130 **converts** the communication path identifier (DA, SA, DSAP, SSAP) in a frame received from the LAN 20 into an ISDN communication. . .

SUMMARY:

BSUM(17)

It . . . an object of the present invention to provide an interconnection system and method capable of facilitating a protocol conversion for **heterogeneous networks**, for example, between the LAN and the ISDN.

SUMMARY:

BSUM(21)

(1) An available ISDN **channel** is determined to **establish** a call with the objective ISDN terminal to open a communication path. In this operation, the DLCI is assigned by the inter-working unit itself or by the ISDN. This results in a correspondence **established** between the communication **path** (DA, SA, DSAP, SSAP) on the LAN side and the

ISDN-side communication path (ISDN channel number, DLI). Consequently, a frame. . .

DRAWING DESC:

DRWD(3)

FIGS. 2A to 2C are diagrams useful to conceptually explain functions of an inter-working unit 100 for **heterogeneous networks** in accordance with the present invention;

DETDESC:

DETD(2)

FIG. 1 shows a network system as an example of a **heterogeneous network** system including an LAN 20 and an ISDN 30. According to the present invention, the network system of FIG. 1. . .

DETDESC:

DETD(11)

FIG. . . . physical layer), an ISDN inter-connector 120 for processing the LAPD-C and the layer 1 (ISDN physical layer), and an LAN/ISDN **protocol converter** 130 for achieving an address translation and a routing control. FIG. 7 shows an example of the constitution of the LAN/ISDN inter-working unit 100 according to the present invention. The LAN/ISDN **protocol converter** 130 includes a memory 132 and a central processing unit (CPU) 131, which are connected to a bus 140. The. . .

DETDESC:

DETD(12)

The . . . 30 will be summarized as follows. The LAN layer 1 (circuit unit 111) monitors a signal on the LAN transmission **path** to **establish** a bit synchronization so as to pass a bit series to the MAC 110. The MAC 110 fetches the received. . .

DETDESC:

DETD(22)

If . . . 512 of the received frame to attain an address DTEi of an objective ISDN terminal. Step 712: An available ISDN **channel** is selected to **establish** a call for the objective ISDN, thereby opening a communication path between the inter-working unit 100 and the ISDN terminal.. . .

CLAIMS:

CLMS(1)

We

including an ISDN channel number and a DLCI assigned to said ISDN communication path thus formed if the ISDN communication **path** is not **established** yet;
converting said received LAN frame to an ISDN frame based on said determined communication path information and sending the ISDN. . .

CLAIMS:

CLMS(10)

10. . . . manner from that used in a communication frame format transmitted in said second network, said method comprising the steps of: **establishing** a first interconnection **path** between a first terminal belonging to the first network and the inter-working unit by communicating a first frame between said. . .

CLAIMS:

CLMS (11)

11. . . .
ISDN terminal or to a reception of a communication frame destined to an ISDN terminal from a LAN terminal for **establishing** an ISDN communication **path** between said inter-work means and said ISDN terminal;

US PAT NO: 5,473,608 [IMAGE AVAILABLE] L22: 7 of 9
TITLE: Method and apparatus for managing and facilitating
communications in a distributed **heterogeneous**
network

ABSTRACT:

A data communication method and apparatus is presented that allows communication in a distributed **heterogeneous network**. Communications managers reside in local processing environments and are responsible for interfacing local end users with the remainder of the **heterogeneous network**. Each communications manager receives distribution units from end users, the distribution units being assigned various priority levels and levels of. . .

SUMMARY:

BSUM(2)

The invention relates to a method and apparatus for managing and facilitating communications in a distributed **heterogeneous network**.

SUMMARY:

BSUM(19)

More particularly, the present invention includes a method and apparatus for communicating between communications managers of a distributed **heterogeneous network**. Within the network, at least one of the communications managers operates on an operating platform which is different from the operating platforms of other communications managers in the **heterogeneous network**.

SUMMARY:

BSUM(20)

To . . . privileged end users and non-privileged end users. The privileged end users typically perform system management functions and communicate through the **heterogeneous network** by use of system management distribution units, whereas non-privileged end users typically transfer information or data distribution units between one. . .

SUMMARY:

BSUM(22)

For each distribution unit, the communications manager determines an adjacent communications manager along a communication path within the **heterogeneous network** from the origin end user to the destination end user. After determining the adjacent communications manager, the distribution unit is. . .

SUMMARY:

BSUM(27)

The . . . processors that share common functions, for example, data

base searching functions. Thus, the information processors which are distributed throughout the **heterogeneous network** can be collected into subsets or complexes of information processors when the information processors within a complex perform common functions.. . .

DRAWING DESC:

DRWD(2)

FIG. 1 is a distributed **heterogeneous network** embodying the communications manager of the present invention.

DETDESC:

DETD(75)

FIGS. . . . Stack specific to that local environment may be used. However, if the application (EU) must communicate with remote applications across **heterogeneous Network** Protocol Stacks then the facilities of the CM of the present invention must be used.

DETDESC:

DETD(324)

If . . . path may range across multiple network protocol stacks. When this occurs, the distribution will arrive at CM Gateways which will **convert** one network **protocol** stack flow to another. Network Nodes which communicate with other Network Nodes via the same network protocol stack are referred. . . .

DETDESC:

DETD(392)

As . . . the EUs. Its function is to provide a standard API to all EUs and to shield the EUs from the **heterogeneous Network** Protocol Stacks within the **heterogeneous network**. In the execution of this functionality the CM serves as the primary interface to the Network Protocol Stack. Therefore, the. . . .

CLAIMS:

CLMS(1)

What is claimed:

1. A method of communicating in a distributed **heterogeneous network** including a plurality of information processors, each information processor having associated therewith an actual communications manager, at least one subset. . . .

CLAIMS:

CLMS(3)

3. A method of communicating in a distributed **heterogeneous network** including a plurality of information processors, each information processor being connected to a network node having associated therewith an actual. . . .

CLAIMS:

CLMS(5)

5. A method of communicating in a distributed **heterogeneous network** including a plurality of information processors, each information processor having associated therewith an actual communications manager, at least one subset. . .

US PAT NO: 5,323,392 [IMAGE AVAILABLE]

L22: 8 of 9

SUMMARY:

BSUM(10)

Because . . . and standards, there is a outstanding need in the telecommunication industry, for communication between data processing devices attached to possibly **heterogeneous networks**. This means interconnecting of these networks, also called internetworking. It is for example easily understandable that the owner of several. . .

SUMMARY:

BSUM(12)

Such . . . protocols but carry on some important differences. Differences are also found at the layer 3 level. Data processing devices and **heterogeneous networks** therefore need some kind of adaptation for interconnecting purpose, and this is achieved by means of adaptation devices. FIG. 4. . .

SUMMARY:

BSUM(28)

converting incoming parameters regarding frame sequence numbering from the first HDLC **protocol**, into **converted** parameters for the second HDLC protocol,

SUMMARY:

BSUM(30)

adding a gap to the value of the last incoming receive parameter from the first **protocol**, to obtain a **converted** send parameter for the first protocol, and